CODING THEORY: A COUNTEREXAMPLE TO G. H. HARDY'S CONCEPTION OF APPLIED MATHEMATICS

NORMAN LEVINSON, Massachusetts Institute of Technology

1. Introduction. A major theme of G. H. Hardy in "A Mathematician's Apology" [5] is the division of mathematics into pure mathematics, "the 'real' mathematics of the 'real' mathematicians which is almost wholly 'useless' " [5, p. 119] and applied mathematics, which he regarded as dull and trivial. In contrast to the harmlessness and innocence of 'real' mathematics, the "trivial mathematics on the other hand has many applications in war." See [5, p. 141]. Hardy exults particularly in the uselessness of number theory which, if "real mathematics" were useful, could be exploited for evil as well as good. Hence "Gauss and lesser mathematicians may be justified in rejoicing that there is one science at any rate, and that their own, whose very remoteness from ordinary human activities should keep it gentle and clean." See [5, p. 121].

Hardy did approve of theoretical physics as exemplified by relativity and quantum mechanics, but regarded them as quite useless [5, p. 135]. If time has shown him wrong about this, it can be argued that in these subjects he was not an expert, and therefore the real test of his ideas concern pure mathematics.

Here we shall show how coding theory refutes Hardy's notion. Finite fields, also called Galois fields, and theorems from number theory play a central role in coding theory. In some areas of applied mathematics, the role of pure mathematics is often at best one of reassurance, such as in providing a nonconstructive existence theorem or a uniqueness theorem, but not in providing the computational or analytic procedures that yield the actual results. In practice the procedures used may involve more intuition and experience than rigor. This is not the case in coding theory, where pure mathematics supplies the constructive procedure for carrying out coding. This may surprise applied mathematicians more than it will pure mathematicians. To accommodate those applied readers, our account will not require familiarity with finite fields or number theory. Rather we shall start with the problem of error correction in the transmission of information by use of codes, and show how this leads to the introduction of a certain mathematical object which is in fact a finite field. Cyclotomic polynomials, a discovery of Gauss, will play a key role.

Quadratic residues and the law of quadratic reciprocity (which Hardy [5, p. 92] regarded as one of the most beautiful theorems of mathematics) also enter

---

Professor Levinson studied with Norbert Wiener and received the Sc.D. degree in 1935. He was a travelling Fellow at Cambridge University in 1934–35 and an NRC Fellow in 1935–37. He has been on the MIT staff since 1937, except for a year as Guggenheim Fellow at the Mathematics Institute, Copenhagen and a year at the University of Tel Aviv. His main research interests are transforms, entire functions, probability, and differential equations. He is the author of the AMS Colloquium volume, *Gap and Density Theorems*, and (with E. Coddington) *Ordinary Differential Equations*. Professor Levinson received the AMS Bôcher Prize and he is a member of the National Academy of Sciences. *Editor.*

coding theory [1, pp. 173, 354]. Another working tool is the Chinese remainder theorem [1, p. 339]. Hardy [5, p. 113] discusses the aesthetic quality of "real mathematics." Here the highly regarded theorems and their proofs possess "a very high degree of *unexpectedness*, combined with *inevitability* and *economy*." This is true of the manner in which finite fields enter coding theory, as we shall see.

Not unexpectedly, finite fields were introduced into coding theory mainly by men trained as mathematicians. Some of the early work was apparently not published. The particular development which will be described here, the BCH codes, is due independently to Bose and Ray-Chauduri [2] and to Hocquenhem [6]. What is most important for the actual usefulness of the method, an efficient decoding process for these codes was discovered by an engineer, Peterson [7]. The BCH codes were generalized considerably by Gorenstein and Zierler [3].

Berlekamp [1, p. vii] states that "the essential limitation of all coding and decoding schemes . . . (has been) the complexity (and *cost*) *of the decoder*. The important work of Reed and Solomon (1960), Bose and Chaudhuri (1960), Gorenstein and Zierler (1961), and Peterson (1961) marked the advent of a new approach to this problem. By associating each digit of certain codes with an element in a Galois field, it was found possible to derive an algebraic equation whose roots represent the locations of the channel errors. . . . As a consequence it is now possible to build algebraic decoders which are orders of magnitude simpler than any that have previously been considered."

The notation used below mainly conforms with that used in Berlekamp [1].

2. Coding. Here a *message* will mean a finite ordered sequence of two symbols which it is desired to transmit through a channel. For example the channel may be a cable or a radio frequency band. It will be convenient to designate the two symbols as 0 and 1. A sequence of k such symbols may be regarded as a *binary k-vector* $(a_1, a_2, \cdots, a_k)$, where each a_j is either 0 or 1. Clearly there are 2^k binary k-vectors. If the transmission channel is noisy, the received vector may differ from the one sent, that is, the transmission process may introduce errors. One way to improve reliability is to repeat the message several times. This is an example of the use of *redundancy*, that is the transmission of more than the k binary digits contained in the original message in order to improve the reliability of the transmission process.

Simple repetition is not efficient. In general, a binary n-vector is transmitted with $n=k+r$, where k is the number of binary digits which form a message and r is the number of redundant digits. These redundant digits are determined according to some rule by the k digits of the message. The process of constructing the redundant n-vector from the message k-vector is called *encoding*. While there are 2^n binary n-vectors, the encoding process leads to a subset of 2^k of these, which may be called *code-vectors*. Because of errors in transmission, the n-vectors which are received need not be code-vectors. The process of correcting the received n-vector and extracting the original k-vector

is called *decoding*. The arithmetic operations in encoding and decoding will be carried out modulo 2, that is, $1+1=0$. This is equivalent to binary addition with no carry-over, and hence is a process easy to design into an electronic computer. The binary arithmetic to be used here need involve only 0 and 1, with rules $0+0=0$, $0+1=1+0=1$, $1+1=0$, $0\cdot 0=0\cdot 1=1\cdot 0=0$ and $1\cdot 1=1$. With these rules, 0 and 1 form a field of two elements, which is known as GF(2) the Galois field of two elements. (This is *not* the place where finite fields play a crucial role in coding theory, since GF(2) by itself is rather trivial.) All arithmetic that follows involving vectors, matrices, and polynomials will be carried out in GF(2). We recall that modulo 2 all even integers may be replaced by 0 and all odd integers by 1.

3. Hamming single error correcting code. Suppose a channel is sufficiently reliable so we can assume that if a binary n-vector is transmitted, then the received binary n-vector contains an error in at most one entry. How much redundancy will allow the position of the error to be determined? Suppose m is a positive integer and set $n=2^m-1$. (This assumption about n here and later is more restrictive than necessary, but is sufficient to illustrate the basic ideas.) A binary number b that could designate which of the n received binary digits contains an error must itself have m digits, because it requires m binary digits to represent the positive integers not exceeding 2^m-1. The occurrence of no error can be designated by all m digits of b zero. As an example, let $m=4$ and hence $n=15$. Then the four-place binary numbers starting with 0001 and ending with 1111 represent all integers from 1 to 15. The above remarks suggest that it may be possible to correct a single error in the transmission of an n-vector, where $n=2^m-1$, if $r=m$ and hence $k=n-m$. A feasible method for doing so was discovered by Hamming [4]. Suppose again that $m=4$ so that $n=15$ and $r=4$; hence $k=11$. All vectors which will be considered from here on will be column vectors which (for typographical reasons) may also be written in row form. Denote the code vectors by $\mathbf{C}=(C_1, C_2, \cdots, C_{15})$, where the C_j are binary digits. Let H be a matrix of 15 columns, each column a binary vector with four entries, all columns distinct, and none identically zero:

$$H=\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \cdots & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & \cdots & 1 \end{pmatrix}.$$

(If a column of H is written as a row, then that row considered as a binary number, represents the column number. This is convenient but not essential.)

Let the eleven entries $C_3, C_5, C_6, C_7, C_9, C_{10}, \cdots, C_{15}$ of $\mathbf{C}$ be the entries of the message k-vector. Determine C_1, C_2, C_4, and C_8 so that

$$H\mathbf{C}=\mathbf{0}. \tag{3.1}$$

This is possible because the square matrix made up of the eighth, fourth, second,

and first columns of H is nonsingular. (Indeed it is the unit matrix.) By (3.1) all code vectors $\boldsymbol{C}$ are orthogonal to the rows of H. Since the arithmetic above is all modulo 2, C_1, C_2, C_4, and C_8 are of course binary digits. This determination of the code vector $\boldsymbol{C}$ completes the encoding process.

Suppose for the moment that errors occur in several of the digits in the transmission of $\boldsymbol{C}$ so that the received binary n-vector $\boldsymbol{R}$ does not coincide with $\boldsymbol{C}$. Performing addition mod 2 componentwise, let the binary n-vector $\boldsymbol{E}$ be defined by

$$\boldsymbol{E} = \boldsymbol{R} - \boldsymbol{C} = \boldsymbol{R} + \boldsymbol{C}. \tag{3.2}$$

If the jth entry of $\boldsymbol{R}$, namely R_j, and that of $\boldsymbol{C}$, namely C_j, coincide, then $E_j = 0$. But if $R_j \neq C_j$, then $E_j = 1$. Hence the vector $\boldsymbol{E}$ has entries differing from 0 at precisely those positions *where an error in transmission occurs*. Consider now

$$H\boldsymbol{R} = H\boldsymbol{C} + H\boldsymbol{E} = H\boldsymbol{E},$$

where use is made of (3.1). If no error has occurred in transmission, then $E = 0$, and so $H\boldsymbol{R} = 0$. If exactly one error has occurred in transmission and this error is in the jth term, then $E_j = 1$ and so

$$H\boldsymbol{R} = H^{(j)},$$

where $H^{(j)}$ is the jth column of H. Clearly knowledge of the vector $H^{(j)}$ determines j. (Indeed with the H above, writing $H^{(j)}$ in row form gives the binary representation of j.) Hence if at most one error occurs, $H\boldsymbol{R}$ determines at which, if any, entry the error occurs. Thus the binary n-vector $\boldsymbol{E}$ is determined, and since $\boldsymbol{C} = \boldsymbol{R} + \boldsymbol{E}$, the vector $\boldsymbol{C}$ is now available. If the entries C_1, C_2, C_4, and C_8 are discarded, the resulting binary k-vector is the original message. This process of reconstructing the message from $\boldsymbol{R}$ is the decoding process. Of course if more than one error occurs, that is, if $\boldsymbol{E}$ has two or more entries which are 1, the above procedure is not valid.

To correct more than one error in $\boldsymbol{R}$, the received binary n-vector, one would expect to increase r, the number of redundant digits, and hence to decrease k. Moreover, for $H\boldsymbol{R}$ then to yield the error locations in $\boldsymbol{R}$, one would expect H to have more rows. The decoding process can be expected to be least complicated if H has a structural pattern based on a reasonably simple mathematical algorithm. A comparatively simple mathematical scheme for locating, at least in principle, up to a prescribed number of errors is used in the BCH codes mentioned earlier. The BCH codes make use of finite fields.

4. The fields GF(2^m). (The reader familiar with finite fields can skim this section.) A column of an m-rowed matrix H may be regarded as a binary m-vector (where of course $m \geqq 1$). If x is an indeterminate and the a_j are all in GF(2), then the polynomial $\sum_0^{m-1} a_j x^j$ of degree $m-1$ can be used to represent a binary m-vector with entries a_j, where $0 \leqq j \leqq m-1$. (Here it is convenient to start the index j at 0.) Since each a_j, for $0 \leqq j \leqq m-1$, can be either 0 or 1, there are a total of 2^m of these polynomials of degree not exceeding $m-1$. Addi-

tion of these polynomials is equivalent to vector addition in GF(2) and leads again to one of the 2^m polynomials.

These rather trivial observations become profound if one further requires that the product of any two of the polynomials is again such a polynomial. Since the degree of the product of two polynomials is the sum of the degrees of each, the degree of the product will be at most $m-1$ only if some artifice is used. One way to achieve this is to compute polynomials modulo a fixed polynomial of degree m which we shall call $f(x)$. Thus if $P(x)$ is a polynomial, then $P(x)$ is equivalent to $P_1(x)$ where $P_1(x)$ is the remainder obtained in dividing $P(x)$ by $f(x)$; thus

$$P(x) = P_1(x) \pmod f$$

if

$$P(x) = J(x)f(x) + P_1(x),$$

where $J(x)$ is a polynomial and the degree of $P_1(x)$ is at most $m-1$. The arithmetic in the division of course is performed in GF(2). In particular $P(x)=0 \pmod f$ if and only if $f(x)$ is a divisor of $P(x)$.

As already stated there are exactly 2^m polynomials of degree not exceeding $m-1$ and with coefficients in GF(2). In this paragraph we shall exclude the null polynomial for which all $a_j=0$. Thus there remain $n=2^m-1$ polynomials. The manipulation of the m-vectors represented by these polynomials becomes particularly simple if the sequence

$$\{x^j\}, \qquad 0 \leqq j \leqq n-1, \pmod f \tag{4.1}$$

generates all n nonnull polynomials.

Example: $m=2$, $f(x)=x^2+x+1$; hence $n=3$. Then the sequence $1, x, x^2, \pmod f$ is $1, x, 1+x$, which are the three nonnull polynomials of degree not exceeding $m-1=1$ with coefficients in GF(2).

We shall show that the sequence (4.1) generates all n of the nonnull polynomials if

$$x^n = 1 \pmod f \quad \text{and} \quad x^k \neq 1 \pmod f \qquad \text{for } 1 \leqq k < n. \tag{4.2}$$

(This is the statement that (4.1) should form a cyclic group of order n.) The x^j, $0 \leqq j \leqq n-1$, are distinct $(\text{mod } f)$. Indeed suppose that

$$x^j = x^k \pmod f, \qquad 0 \leqq j < k \leqq n-1.$$

Then multiply the above equation by x^{n-k} and use the first equation of (4.2) to get

$$x^{n-(k-j)} = 1 \pmod f.$$

Since $k>j$ this violates (4.2). Furthermore no $x^j=0 \pmod f$, since multiplying by x^{n-j}, we should have $1=0 \pmod f$ or $f(x)$ divides 1, which is impossible since degree $f=m \geqq 1$. Hence the sequence (4.1) of n elements are all distinct $(\text{mod } f)$

and none is the null element. Therefore the sequence (4.1) generates all n of the nonnull polynomials if (4.2) holds.

If $y=x^i$ for some fixed $i\geqq 1$, then the least positive integer λ for which $y^\lambda=1 \pmod f$ is called the *order* of y. If $y^k=1 \pmod f$ for some $k\geqq 1$, then k is a multiple of λ. Indeed let $k=q\lambda+s$ where $q\geqq 0$ and $0\leqq s<\lambda$. Then $1=y^k=y^{\lambda q+s}=y^s \pmod f$. From the definition of λ, since $s<\lambda$, this implies $s=0$ and proves the following special case of a classical result:

LEMMA 4.1. *Let y be a power of x and let y be of order λ. If $y^k\equiv 1 \pmod f$, then k is a multiple of λ.*

From (4.2), $f(x)$ must be a factor of x^n-1. Let us now again take the case $m=4$ (so that $n=15$) and enumerate certain particularly relevant factors of $x^{15}-1$. We revert to ordinary arithmetic, and note that if $x^3=1$ or if $x^5=1$, then certainly $x^{15}=1$. Hence x^3-1 and x^5-1 are factors of $x^{15}-1$. Of course $x-1$ is a factor of all of these. We now write the obvious identity

$$\begin{aligned} x^{15}-1 &= (x-1)\frac{x^3-1}{x-1}\,\frac{x^5-1}{x-1}\left(\frac{(x^{15}-1)(x-1)}{(x^3-1)(x^5-1)}\right) \\ &= Q^{(1)}(x)Q^{(3)}(x)Q^{(5)}(x)Q^{(15)}(x), \end{aligned} \tag{4.3}$$

where

$$Q^{(1)}(x)=x-1, \qquad Q^{(3)}(x)=x^2+x+1, \qquad Q^{(5)}(x)=x^4+x^3+x^2+x+1,$$

and (as can be verified) $Q^{(15)}(x)=x^8-x^7+x^5-x^4+x^3-x+1$.

REMARK: The polynomials $Q^{(j)}(x)$ above are examples of the cyclotomic polynomials of Gauss, Eisenstein, etc. A root of unity ρ is said to have order $j\geqq 1$ if j is the least exponent for which $\rho^j=1$. As in Lemma 4.1, the roots of $x^{15}-1$ must all have orders which are factors of 15. The above factorization (4.3) involves the roots of orders 1, 3, 5, and 15, and these occur precisely in $Q^{(1)}$, $Q^{(3)}$, $Q^{(5)}$, and $Q^{(15)}$ respectively. The polynomial $Q^{(j)}(x)$ has roots all of which are of order j.

REMARK: The choice $m=4$ is not entirely an accident. Note that the cases $m=3$ and $m=5$ would not serve nearly as well as illustrative examples because 7 and 31 are prime numbers, and so the analogue of (4.3) would be too simple to be revealing. The case $m=6$ becomes computationally rather long to serve as a suitable example.

Return again to GF(2); there $Q^{(1)}$, $Q^{(3)}$, and $Q^{(5)}$ remain the same in (4.3). But as can be readily verified,

$$Q^{(15)}(x)=(x^4+x^3+1)(x^4+x+1).$$

Now take $f(x)$ as one of the two quartic factors of $Q^{(15)}$, say

$$f(x)=x^4+x^3+1. \tag{4.4}$$

Then since $f(x)$ is a factor of $Q^{(15)}(x)$, it is a factor of $x^{15}-1$; therefore $x^{15}=1$

(mod f). By Lemma 4.1, the order of x must be a divisor of 15, so it is 1, 3, 5, or 15. But $f(x)$ is obviously not a divisor of $x-1$ or x^3-1 and can easily be shown not to be a divisor of x^5-1. Hence the order of x is not 1, 3, or 5, so it must be 15. (This is in fact a particular instance of an easily proved general property of the cyclotomic polynomials.) Therefore (4.2) is satisfied, so 1, x, x^2, $\cdots$, x^{14} (mod f) are the 15 cubic polynomials with coefficients in GF(2), none of which is the null polynomial. Given any x^j, $1 \leqq j \leqq 14$, then x^{15-j} is obviously its inverse (mod f). Thus these polynomials form a group under multiplication (mod f). (It is of course the cyclic group.) If the null polynomial is adjoined, then the 16 polynomials obviously form a group under addition. It follows readily that mod f these 16 cubic polynomials with coefficients in GF(2) form a field of 16 elements. This field is known as GF(16).

REMARK: The polynomial (4.4) is irreducible, that is, it cannot be written as the product of two lower degree polynomials in the arithmetic of GF(2). Indeed if it could, we should have

$$f_1(x)f_2(x) = f(x).$$

But $f_1(x)$ is a member of GF(16), hence has an inverse; the same is true of $f_2(x)$. If we multiply by these, we obtain $1=0$ (mod f), which is impossible.

REMARK: In principle the entire above procedure can be carried out to establish the existence of GF(2^m) for any m, and to specify an appropriate $f(x)$ of degree m. Actually to treat the general case it is necessary to develop a little more theory concerning $Q^{(j)}(x)$ and the irreducible polynomials with coefficients in GF(2), [1].

A more convenient way to indicate that we are working mod $f(x)$ is to let α denote a root of $f(x)$. Therefore $\alpha^4+\alpha^3+1=0$ and so any polynomial in α is automatically equivalent to a binary cubic in α. It is the cubic which one computes working mod f since the only property of α that is used is $f(\alpha)=0$. Thus the elements of GF(16) may be designated by the binary cubics in α.

Summary: Let α be a root of $\alpha^4+\alpha^3+1=0$. (Only this equation, and not the actual numerical value of α, is used.) Then each polynomial in α with coefficients in GF(2) is equal to a binary cubic polynomial in α. There are $2^4=16$ binary cubics. These form the field GF(16). Moreover $\{\alpha^j\}$, for $0 \leqq j \leqq 14$, generates the 15 nonnull binary cubics which together with the null polynomial make up GF(16). A binary cubic may be viewed as a binary 4-vector.

5. A multiple error correcting code. To show how finite fields enter into coding, let us continue with the case $m=4$, $2^m=16$. Suppose now it is desired to correct up to 3 errors in the transmission of the encoded vector $\boldsymbol{C}$ with $n=15$ entries. Since with $n=15$ the correction of one error required a 4-rowed matrix, as displayed above (3.1), it seems plausible to try to correct three errors with a 12-rowed matrix. This operating on $\boldsymbol{R}$ leads to a 12-vector which can be viewed as three 4-vectors and hence contains sufficient information to determine three integers between 1 and 15 and thereby locate up to three errors in $\boldsymbol{R}$. A systematic way to construct H is with its twelve rows arranged in three blocks of

four rows as follows:

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^{14} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \cdots & \alpha^{42} \\ 1 & \alpha^5 & \alpha^{10} & \alpha^{15} & \cdots & \alpha^{70} \end{bmatrix}.$$

Each power of α of course represents a binary 4-vector belonging to GF(16). Why the row blocks α^{2j} and α^{4j}, $0 \leqq j \leqq 14$, can be omitted will soon be apparent. (Since $\alpha^4 = \alpha^3 + 1$ in GF(2), the first block of four rows can be computed from $1\ \alpha\ \alpha^2 \cdots \alpha^{14}$ and is

$$\begin{array}{cccccccccccccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0. \end{array}$$

The second block of four rows of H, namely $1\ \alpha^3\ \alpha^6 \cdots \alpha^{42}$, consists of the first, fourth, seventh, tenth, and thirteenth columns of the above displayed matrix repeated three times. The third block of four rows consists of the first, sixth, and eleventh columns of the above displayed matrix repeated five times.)

The received binary 15-vector $\boldsymbol{R}$ with entries R_j, $0 \leqq j \leqq 14$, has the polynomial representation

$$R(x) = \sum_0^{14} R_j x^j.$$

(Here we are *not* computing $R(x) \bmod f$.) The application $H\boldsymbol{R}$ of H to $\boldsymbol{R}$ is the column vector with 12 entries

$$R(\alpha),\ R(\alpha^3),\ R(\alpha^5)$$

which is represented in row form as a triple of 4-vectors, each 4-vector being an element of GF(16).

As in (3.2), let $\boldsymbol{R} = \boldsymbol{C} + \boldsymbol{E}$. If $\boldsymbol{E} = 0$, then it is desirable that $H\boldsymbol{R} = 0$ and therefore we should have $H\boldsymbol{C} = 0$. Let

$$C(x) = \sum_0^{14} C_j x^j.$$

In terms of α, $H\boldsymbol{C}$ is the binary 12-vector $C(\alpha)$, $C(\alpha^3)$, $C(\alpha^5)$. For this 12-vector to be zero, the polynomial $C(x)$ of degree 14 should vanish for $x = \alpha$, α^3, and α^5. To make $C(x)$ vanish for $x = \alpha$, we can require that $f(x)$ be a factor of $C(x)$ in GF(2). It will be convenient now to designate $f(x)$ by $M_1(x)$. To find a polynomial, say $M_3(x)$, which has α^3 as a root, note that α^3 has order 5 and hence will be a root of $Q^{(5)}(x)$ of (4.3), where

$$Q^{(5)}(x) = 1 + x + x^2 + x^3 + x^4.$$

Denote $Q^{(5)}(x)$ by $M_3(x)$. Similarly α^5 has order 3 and hence is a root of $Q^{(3)}(x) = 1+x+x^2$, which we shall denote by $M_5(x)$. (The polynomials M_1, M_3, and M_5 are all minimum polynomials in the sense that no polynomials of lower degree with coefficients in GF(2) have α, α^3, and α^5 respectively as roots.) Let

$$g(x) = M_1(x)M_3(x)M_5(x).$$

Then the degree of $g(x)$ is 10, since M_1 and M_3 are of a degree 4 and M_5 of degree 2. Moreover $g(x)$ vanishes for $x=\alpha$, α^3, and α^5 because $M_1(\alpha)$, $M_3(\alpha^3)$, and $M_5(\alpha^5)$ are each zero. We now require that $g(x)$ be a factor of $C(x)$ to assure that $C(x)$ has α, α^3, and α^5 as roots. Recall that $\boldsymbol{C}$ is a vector with 15 entries. Let C_{10}, C_{11}, C_{12}, C_{13}, C_{14} be a message vector of $k=5$ binary digits. Choose $\sum_0^9 C_j x^j$ as the negative of the remainder of the quotient

$$\frac{C_{14}x^{14} + C_{13}x^{13} + \cdots + C_{10}x^{10}}{g(x)}$$

so $C(x)$ will indeed have $g(x)$ as a factor and hence α, α^3, and α^5 as roots. The above arithmetic is of course in GF(2). This is the encoding process with $n=15$, $k=5$, and $r=10$. The binary polynomial $g(x)$ is known as the *generator polynomial* of the code. Indeed a code-vector $\boldsymbol{C}$ is characterized by the fact that $C(x)$ is divisible by $g(x)$.

It will now be shown in principle at least that the 12-vector $\boldsymbol{HR}$ can be used to correct up to a maximum of 3 errors in transmission. Since $\boldsymbol{HC}=0$, therefore $\boldsymbol{HR}=\boldsymbol{HE}$, and the 12 vector $\boldsymbol{HE}$ regarded as a triple of 4-vectors determines $E(\alpha)$, $E(\alpha^3)$, and $E(\alpha^5)$. We recall that the entries of $\boldsymbol{E}$ are 1 where an error occurs and 0 otherwise. Suppose 3 errors occur say at the entries i_1, i_2, and i_3 of $\boldsymbol{E}$. Then

$$\begin{aligned} E(\alpha) &= \alpha^{i_1} + \alpha^{i_2} + \alpha^{i_3}, \\ E(\alpha^3) &= \alpha^{3i_1} + \alpha^{3i_2} + \alpha^{3i_3}, \\ E(\alpha^5) &= \alpha^{5i_1} + \alpha^{5i_2} + \alpha^{5i_3}. \end{aligned}$$

It will be convenient to note that with the a_j in GF(2),

$$(\sum a_j\alpha^j)^2 = \sum a_j^2\alpha^{2j} = \sum a_j\alpha^{2j}, \tag{5.1}$$

because all cross products have $2=1+1$ as a factor. Suppose now that 3 errors occur at the positions i_4, i_5, and i_6, all distinct from i_1, i_2, and i_3 above, and suppose these errors lead to the same values for $E(\alpha)$, $E(\alpha^3)$, and $E(\alpha^5)$ as do i_1, i_2, and i_3. Then this leads to

$$\alpha^{ji_1} + \alpha^{ji_2} + \alpha^{ji_3} = \alpha^{ji_4} + \alpha^{ji_5} + \alpha^{ji_6}, \qquad j = 1, 3, 5$$

or

$$\sum_{d=1}^{6} \alpha^{ji_d} = 0 \qquad j = 1, 3, 5. \tag{5.2}$$

But now applying (5.1) to the case $j=1$, (5.2) holds for $j=2$. Applying (5.1) to the case $j=2$ and then to $j=3$ gives (5.2) for $j=4$ and $j=6$. (That is why we omitted the even powers of α from the rows of H.) Thus

$$\sum_{d=1}^{6} \alpha^{ji_d} = 0 \qquad j = 1, 2, 3, 4, 5, 6. \tag{5.3}$$

The determinant of the above system is a Vandermonde determinant equal to

$$\alpha^{i_1+i_2+\cdots+i_6} \prod_{6 \geq d > e \geq 1} (\alpha^{i_d} - \alpha^{i_e}). \tag{5.4}$$

Each factor of (5.4) is

$$\alpha^{i_d} - \alpha^{i_e} = \alpha^{i_e}(\alpha^{i_d - i_e} - 1).$$

Since $0 \leqq i_d \leqq 14$, it follows that $0 < |i_d - i_e| \leqq 14$. Hence, since the order of α is 15, no factor of (5.4) is zero so the determinant is not zero. Thus the homogeneous system (5.3) is impossible. If the other cases—such as two sets of three errors but with some in common, or where one set or both sets have less than three errors—are considered, there are now fewer columns in the analogue to (5.3); hence some rows may be discarded leading again to a Vandermonde situation. Thus if there are at most three errors, then $H\boldsymbol{R}$ determines their locations uniquely.

Of course for successful decoding, the above uniqueness result, while reassuring, must be replaced with a reasonably simple constructive procedure for determining which entries of the vector $\boldsymbol{E}$, if any, are 1. It is the simple orderly structure of H in terms of powers of α that makes the mechanization of such a decoding procedure feasible [1, Chap. 7].

The BCH codes, based on conceptions from pure mathematics, are not unique in using unexpected parts of pure mathematics for coding. Among the codes there are for example euclidean-geometry codes [1, p. 375], projective-geometry codes [1, p. 376], tensor product codes [1, p. 346], and quadratic residue codes [1, p. 354].

The preparation of this paper was supported in part by the Office of Naval Research and by the National Science Foundation NSF GP-13778.

References

1. Elwyn R. Berlekamp, Algebraic Coding Theory, McGraw-Hill, New York, 1968.
2. R. C. Bose and D. K. Ray-Chaudhuri, On a class of error correcting binary group codes, Information and Control, 3 (1960) 68–79 and 279–290.
3. D. C. Gorenstein and N. Zierler, A class of error-correcting codes in p^m symbols, J. Soc. Indust. Appl. Math., 9 (1961) 207–214.
4. R. W. Hamming, Error detecting and error correcting codes, Bell System Tech. J., 29 (1950) 147–160.
5. G. H. Hardy, A Mathematician's Apology, Cambridge University Press, 1967 Edition.
6. A. Hocquenhem, Codes correcteurs d'erreurs, Chiffres, 2 (1959) 147–156.
7. W. W. Peterson, Error-correcting Codes, M.I.T. Press, 1961.
8. I. A. Reed and G. Solomon, Polynomial codes over certain finite fields, J. Soc. Indust. Appl. Math., 8 (1960) 300–304.